High-Capacity Image Steganography Based on Side Match Block Matching

Jaeyoung Kim and Hanhoon Park
Department of Electronic Engineering, Pukyong National University, Busan, Republic of Korea.
hanhoon_park@pknu.ac.kr

***Abstract*—In this paper, we proposed an image steganography method which exploits a variant of block matching (BM) algorithm, called side match block matching (SMBM), and discrete wavelet transform (DWT) for embedding high-capacity images. The proposed method can greatly improve the embedding capacity by using SMBM. In addition, the proposed method resolves the problem with the loss of least significant bits in previous BM-based steganography methods and improves the confidentiality by embedding secret information in the DWT high frequency regions.**

***Index Terms*—Image Steganography; Block Matching; Discrete Wavelet Transform; High-Capacity; Side Match Block Matching.**

I. INTRODUCTION

Steganography is a technology for covert communication where a secret message is transferred/received while not being perceived by the third person between a receiver and a transmitter. It has been exploited for thousands of years to communicate military, diplomatic or business information. However, the recent development of digital information and communication technology has enabled the development of more various steganography methods to exchange or protect confidential information in more various fields, including medical privacy and copyright management [1].

Image steganography is one of the digital steganography methods that use images as carriers (usually called cover images). Image steganography can hide secret information in a spatial domain or a transform domain. The well-known information hiding methods in the spatial domain are the least significant bit (LSB) [2] and the pixel value differencing (PVD) [3] algorithms. The LSB is a method that replaces *B* less significant bits per a pixel of a cover image with secret bits. The PVD is a method that embeds secret information by modulating the difference between neighboring pixels. The information hiding methods in the transform domain transforms the cover image using discrete wavelet transform (DWT) [4] or discrete cosine transforms (DCT) [5] and usually embeds secret information into the transform coefficients.

In this paper, we proposed a high-capacity image steganography method that transforms the cover image using DWT and embeds secret images into the high frequency regions using a block matching (BM) algorithm. The BM-based embedding method in the DWT domain can resolve the problem with the loss of LSB bits in the BM-based embedding method in the spatial domain [6]. To further improve the embedding capacity, we modified the BM algorithm to a side match BM (SMBM) with an assumption of the high correlation between neighboring blocks, which was inspired by side match vector quantization (SMVQ) [7].

II. RELATED WORKS

A. Steganography Using Block Matching

In general, the BM-based steganography is a method that hides secret information by finding and matching similar blocks that minimize the Euclidean distance between secret and cover blocks [6]. The Euclidean distance between two $N \times N$ blocks $\mathbf{A}$ and $\mathbf{B}$ is given by Equation (1).

$$d(\mathbf{A},\mathbf{B}) = \sum_{m=1}^{N}\sum_{n=1}^{N}\big(\mathbf{A}(m,n) - \mathbf{B}(m,n)\big)^2 \qquad (1)$$

The cover image and the secret image are divided into non-overlapping blocks, and for each secret block, a cover block minimizing the Euclidian distance is searched for all the blocks of the cover image. The coordinates or index of this block become the final address information and are embedded into LSBs of the cover image, and the secret image is restored from the stego image (i.e., the cover image after embedding the secret information) through this address information in the extraction process.

B. Discrete Wavelet Transform

When the wavelet transform is applied to an image [4], the image is divided into the LL sub-band, which contains low frequency components easily recognizable by HVS (human visual system), and the HL, LH, and HH sub-bands, which contain high frequency components. Letting $s_{i,l}$ be the *l*-th pixel of the *i*-th level of a wavelet-transformed image, then wavelet transform can be expressed by Equation (2).

$$s_{i,l} = (s_{i-1,2l+1} + s_{i-1,2l})/2,\ d_{i,l} = s_{i-1,2l+1} - s_{i-1,2l} \qquad (2)$$

Here, $s_{i,l}$ and $d_{i,l}$ are the low and high frequency coefficients, respectively. The transformation in (2) is called DWT. In DWT, however, the fractional part of the real number is lost and thus the original image information can be lost. Thus, an integer wavelet transform (IWT) is used to avoid the image information loss and is expressed by Equation (3).

$$d_{i,l} = s_{i-1,2l+1} - s_{i-1,2l},\ s_{i,l} = s_{i-1,2l} + \lfloor d_{i,l}/2 \rfloor \qquad (3)$$

In IWT, after calculating the high frequency components, the low frequency components are calculated. The inverse transformation is shown in Equation (4).

$$s_{i-1,2l} = s_{i,l} + \lfloor d_{i,l}/2 \rfloor, s_{i-1,2l+1} = s_{i-1,2l} + d_{i,l} \quad (4)$$

III. PROPOSED METHOD

In this paper, we proposed an image steganography method that embeds high-capacity secret image in the IWT domain of cover image using SMBM algorithm. Generally, in the case of using the spatial BM algorithm [6], the whole cover image becomes a searching space for block matching and its LSBs are modified by information embedding. Therefore, the searching space for block matching can be different between in the embedding phase and in the extraction phase. To address this, IWT is used to classify the LL sub-band, which is the searching space for block matching, and the LH, HL and HH sub-bands, which are the embedding spaces.

A. Side Match Block Matching

The SMBM algorithm first applies the BM algorithm to the top and leftmost blocks of the secret image to get the matched cover blocks (see Figure 1). Then, for each remaining block in a scanline order, it generates a subset of cover blocks (hereafter named codebooks) with side pixels similar to those obtained from the matched cover blocks of the upper and left neighboring blocks (see Figure 2), and find a matched cover block from the codebook. Therefore, the number of bits required to embed a block of the secret image is reduced and higher-capacity secret image can be embedded.

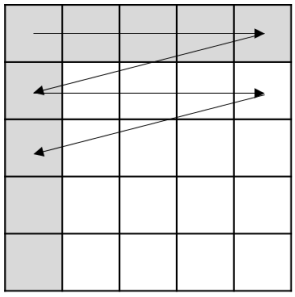

Figure 1: Blocks with different embedding algorithms and embedding order. Shaded blocks: using BM, other blocks: using SMBM

To explain the SMBM algorithm in more details, several symbols are defined as shown in Table 1.

Table 1
Definition of the Symbols

Symbol	Definition
$N \times N$	Block size
N_C	Size of the codebook
CB	A block in the codebook
$\mathbf{S}_{(sy,sx)}(m,n)$	A divided non-overlapping block of the secret image
$\mathbf{C}_{(sy,sx)}(m,n)$	A divided non-overlapping block of the cover image
$\mathbf{SB}_{(sy,sx)}(m,n)$	A matched non-overlapping block of the secret image

Side vectors, **M**1 and **M**2, are generated with a length of 2*N*-1 by referring to the side pixels (gray region and yellow region, respectively) as shown in Figure 2, from the current block of the secret image and the matched cover blocks of its upper and left neighboring blocks for generating the codebook as shown in (5).

$$\begin{aligned}
&\mathbf{M}1_{(sy,sx)}(l) \\
&= (\mathbf{SB}_{(sy,sx-1)}(1,N) + \mathbf{SB}_{(sy-1,sx)}(1,N))/2, \ \ (l=1), \\
&= \mathbf{SB}_{(sy,sx-1)}(1,l-1), (2 \le l \le N), \\
&= \mathbf{SB}_{(sy-1,sx)}(l-(N+1),1), \ (N+1 \le l \le 2N-1), \\
&\mathbf{M}2_{(cy,cx)}(l) \\
&= \mathbf{C}_{(cy,cx)}(1,l-1), \ (1 \le l \le N), \\
&= \mathbf{C}_{(cy,cx)}(l-(N+1),1), \quad (N+1 \le l \le 2N-1).
\end{aligned} \quad (5)$$

The set of a fixed number of cover blocks whose **M**2 vectors have small Euclidean distances to **M**1 vector is used as a codebook for embedding an **S** block. Sorting is performed on the generated codebook. Then, for all **CB**s of the codebook, we find CB_{index} with the smallest Euclidean distance for all the pixels with S, and this index is the address information.

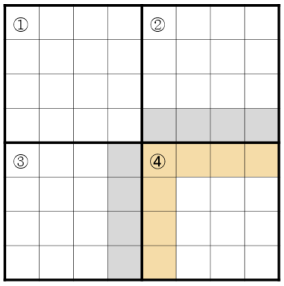

Figure 2: Side vector generation from already-matched upper and left blocks

When using the SMBM algorithm, the problem shown in Figure 3 may occur in blocks, including strong edges or blocks whose neighboring blocks have too different texture. Since the codebook is generated by matching the side pixels only, an error occurs when there are many high frequency components such as edges or noise pixels in the block. If the error is large, it cannot match an appropriate block, and the error is accumulated in the process of matching adjacent blocks. To overcome these problems, the secret blocks are classified in advance into two groups: edge blocks and non-edge blocks. To identify each block, the Canny edge detector is used and the number of edge pixels is counted. Then, for edge blocks, the original BM algorithm is applied. Only for non-edge blocks, the SMBM algorithm is applied. Consequently, for each secret block, an additional bit for identifying whether it is edge block or not is embedded and used in the extraction phase.

B. Embedding Procedure of the Proposed Method

The embedding procedure consists of three steps. First, IWT is applied to the cover image. Then, the LL bands of the cover image and the secret image are divided into non-overlapping blocks. Next, by applying BM or SMBM to each secret block, the location map and address map are generated. Finally, the generated location map and address map are converted into a bitstream and embedded into the *B*-LSB of the LH, HL and HH sub-bands of the cover image. A stego image is obtained by applying inverse IWT to the resulting sub-bands of the cover image.

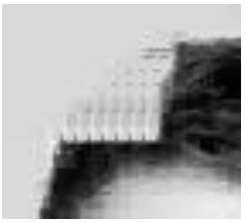

Figure 3: Error accumulation of the SMBM algorithm

When the secret image size is $S_w \times S_h$, the location map is generated as follows. An empty location map and an empty address map with a size of $\lfloor S_h/N \times S_w/N \rfloor$ are created. Each pixel of the location map corresponds to whether each block of the secret image is edge block or not. Write '1' for the edge blocks and '0' for the non-edge blocks. Based on the generated location map, after deciding whether applying BM or SMBM, an address map is generated by writing the resulting indices. The number of bits required for saving each index resulting from BM and SMBM is computed as in Equation (6).

$$b_{\mathrm{BM}} = \log_2(\lfloor C_h/2N \rfloor \times \lfloor C_w/2N \rfloor), \quad b_{\mathrm{SMBM}} = \log_2(\lceil N_C \rceil) \tag{6}$$

Here, C_w and C_h are the width and height of the cover image.

C. Extracting Procedure of the Proposed Method

In order to extract the secret image, the number of bit planes used for embedding the secret bitstream *B*, the block size $N \times N$, the secret image size $S_w \times S_h$, and the codebook size N_C are required. First, IWT is applied to the stego image and extract the bitstream from the *B*-LSB of the LH, HL and HH sub-bands of the stego image. Then, the location map and address map are obtained from $\lfloor S_h/N \rfloor \times \lfloor S_w/N \rfloor$ bits in the front part of the extracted bitstream and the remaining bits, respectively. Finally, if the location map bit is '1', the index with b_{BM} bits is extracted from the address map and the associated block is found/restored from the LL sub-band of the stego image. If the location map bit is '0', the index with b_{SMBM} bits is extracted from the address map. Then, the codebook is generated in the same way as at the embedding phase from the LL sub-band of the stego image and the block associated with the index is found/restored from the codebook.

IV. Experiment and Results

The experiment uses two 1-channel 8 bits grayscale images, lena.bmp and Einstein.bmp (see Figure 4). The einstein.bmp was embedded into 2-LSB of the lena.bmp. The embedding capacity and the peak signal-to-noise ratio (PSNR) according to the codebook size and the block size were analyzed. The embedding capacity was quantified by the value of bits per pixel (*bpp*), which indicates the number of index bits required to embed a secret pixel. The lower the value of *bpp* is, the better the embedding capacity is. If the number of blocks to which the BM algorithm is applied is *NB* and the number of blocks of the secret image is *NS*, the required bitstream to embed the secret image is expressed by (7).

$$bpp = \frac{NS + NB \times b_{BM} + (NS - NB) \times b_{SMBM}}{S_w \times S_h} \tag{7}$$

where $NS = \lfloor S_h/N \rfloor \times \lfloor S_w/N \rfloor$

For example, if the size of the secret and cover image is 512×512, the codebook size is 64 and the block size is 8×8, the *bpp* is as follows.

$$\frac{64 \times 64 + 1000 \times 10 + (4096 - 1000) \times 6}{512 \times 512} = 0.125$$

That is, 0.125 bits are required to embed a pixel of the secret image. PSNR is used for quantifying the ability of concealing and restoring the secret image by comparing between the cover image and the stego image, between the secret image and the extracted image.

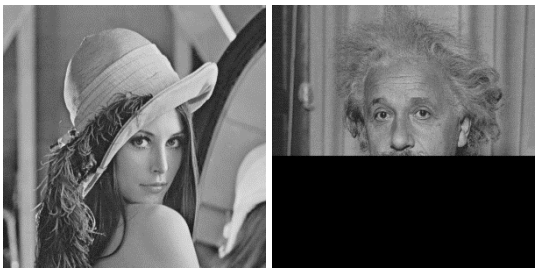

Figure 4: The cover image (lena.bmp) and secret image (einstein.bmp)

A. Embedding Capacity and PSNR According to the Codebook Size

The block size was fixed to 4×4. The codebook size was changed to 32, 64, 128, 256, and 512. The Canny edge detector was used to identify the blocks to apply the BM or SMBM algorithm (see Figure 5).

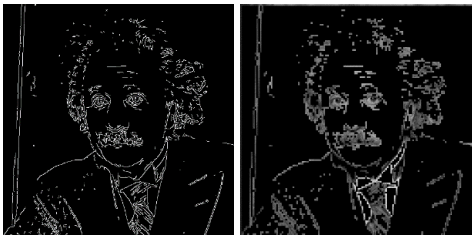

Figure 5: Edge map and stego blocks with BM algorithm applied

Table 2
Comparison of the PSNRs and the Embedding Capacity Between BM Only and the Proposed Method According to the Codebook Size

	BM only	Proposed method				
Codebook size	-	32	64	128	256	512
Secret PSNR	32.053	30.609	31.012	31.504	31.821	31.985
Stego PSNR	47.606	49.778	49.291	48.910	48.495	48.179
bpp	0.75	0.449	0.501	0.553	0.605	0.657

The experimental results are shown in Table 2 and Figure 6. "BM only" indicates that the BM algorithm is applied to all the secret blocks and the SMBM algorithm is not used. The value of *bpp* for the proposed method was always lower than that of the BM only and decreased by decreasing the codebook size. When the codebook size was 32, the embedding capacity was increased by 70% with a little loss in PSNR. The loss in PSNR caused the loss of high details in the secret image as shown in Figure 7. However, the visual difference was not noticeable.

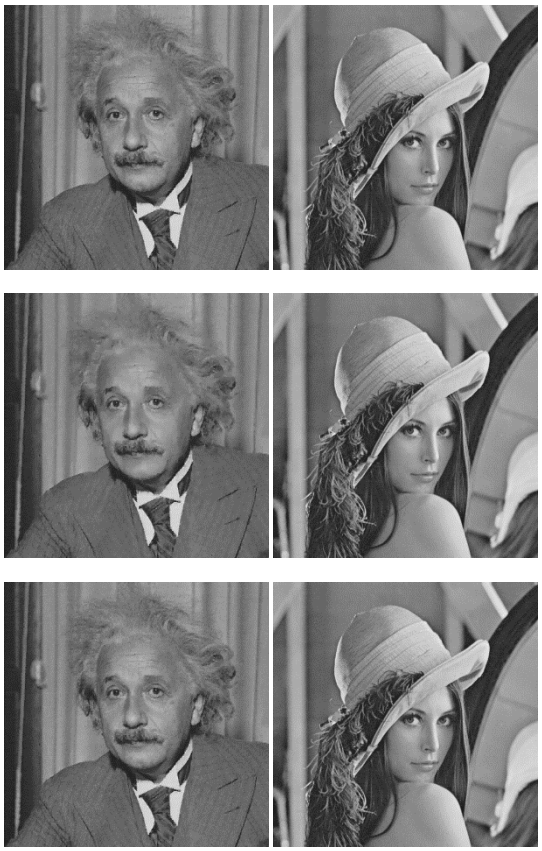

Figure 6: The extracted secret images (left) and stego images (right). First row: BM only, second and third row: proposed method with the codebook size of 32 and 512, respectively

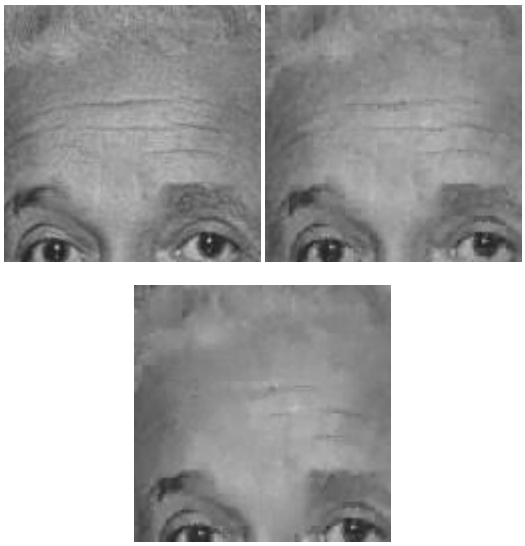

Figure 7: Enlargement of the original secret image and extracted secret images in Figure 4 and Figure 6

B. Embedding Capacity and PSNR According to the Block Size

The codebook size was fixed to 64. The block size was changed to 4×4, 6×6 and 8×8. The experimental results are shown in Table 3. For both the BM only and the proposed method, the embedding capacity increased in proportion to the block size. However, the PSNR decreased in proportion to the block size. The smaller the block size was, the more improved the embedding capacity was. The improvement was 49.8%, 28.8% and 25.6% when the block size was 4×4, 6×6 and 8×8, respectively. Regarding the visual quality of stego images, the PSNR of the proposed method was better because the amount of modulation in the cover image required to embed the same sized image was smaller.

Table 3
Comparison of the PSNRs and the Embedding Capacity Between BM Only and the Proposed Method According to the Block Size

	BM only			Proposed method		
Block size	4×4	6×6	8×8	4×4	6×6	8×8
Secret PSNR	32.056	29.370	27.148	31.113	29.044	25.626
Stego PSNR	47.606	50.882	54.355	49.291	51.901	55.308
bpp	0.75	0.366	0.156	0.501	0.284	0.124

V. CONCLUSION

In this paper, a high-capacity image steganography method was proposed. It embedded secret images in the DWT domain using SMBM. The proposed method could achieve higher embedding capacity than using BM with a little loss in the visual quality of secret image. Future work will focus on further improving the PSNR of the extracted secret image while maintaining the embedding capacity.

ACKNOWLEDGEMENT

This work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.

REFERENCES

[1] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: survey and analyses of current methods," *Sig. Proc.*, vol. 90, no. 3, pp. 727-752, 2010.

[2] C.-K. Chan and L.M. Cheng, "Hiding data in images by simple LSB substitution," *Patt. Recog.*, vol. 37, no. 3, pp. 469-474, 2004.

[3] D.C. Wu and W.H. Tsai, "A steganographic method for images by pixel-value differencing," *Patt. Recog. Lett.*, vol. 24, pp. 1613-1626, 2003.

[4] R.O. El Safy, H.H. Zayed, and A. El Dessouki, "An adaptive steganographic technique based on integer wavelet transform," in *Proc. Int. Conf. Netw. and Media Conv.*, 2009, pp. 111–117.

[5] T. Bhattacharya, N. Dey, and S.R.B. Chaudhuri, "A session based multiple image hiding technique using DWT and DCT," *Int. J. Comp. Appl.*, vol. 38, no. 5, pp. 398–409, 2012.

[6] R.-Z. Wang and Y.-D. Tsai, "An image-hiding method with high hiding capacity based on best-block matching and k-means clustering," *Patt. Recog.*, vol. 40, no. 2, pp. 398-409, 2007.

[7] T. Kim, "Side match and overlap match vector quantizers for images," *IEEE Trans. Image Proc.*, vol. 1, no. 4, pp. 170–185, 1992.